When should we prefer Decision Transformers for offline Reinforcement Learning?

Prajjwal Bhargava, Rohan Chitnis, Alborz Geramifard, Shagun Sodhani, Amy Zhang

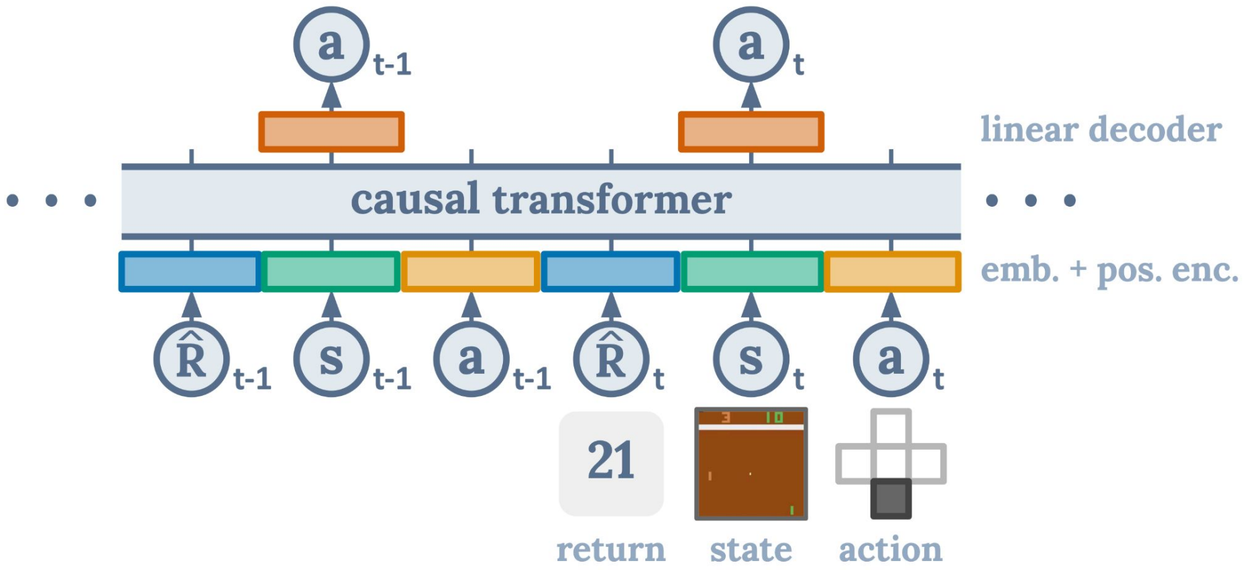
International Conference on Learning Representations (ICLR) 2024

Meta

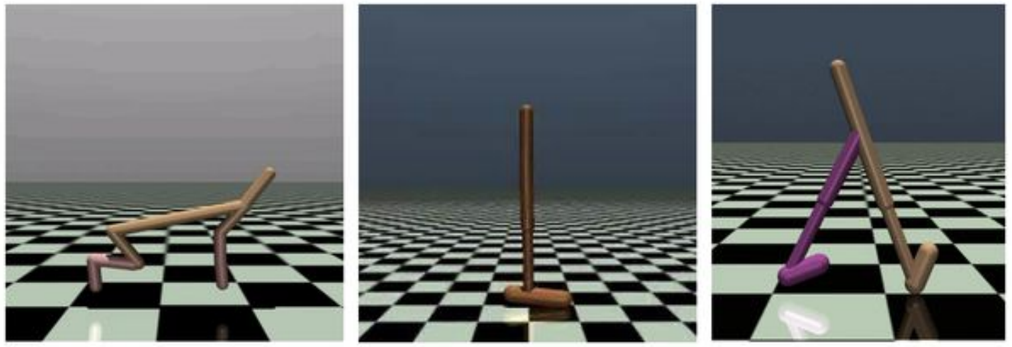
Motivation

Which learning method is preferred for offline reinforcement learning?

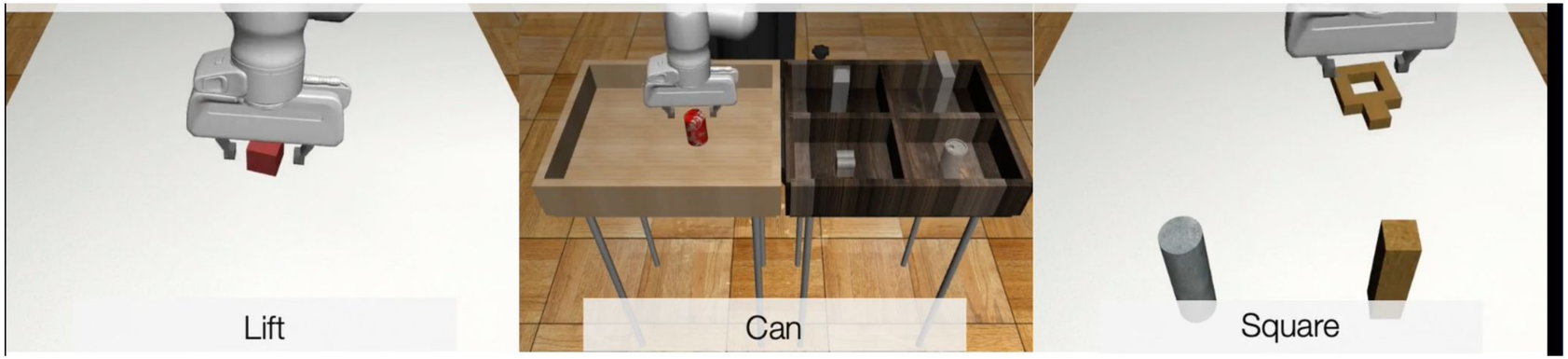
- ➔ Conservative Q-Learning (CQL)
- ➔ Behaviour Cloning (BC)
- ➔ Decision Transformers (DT)



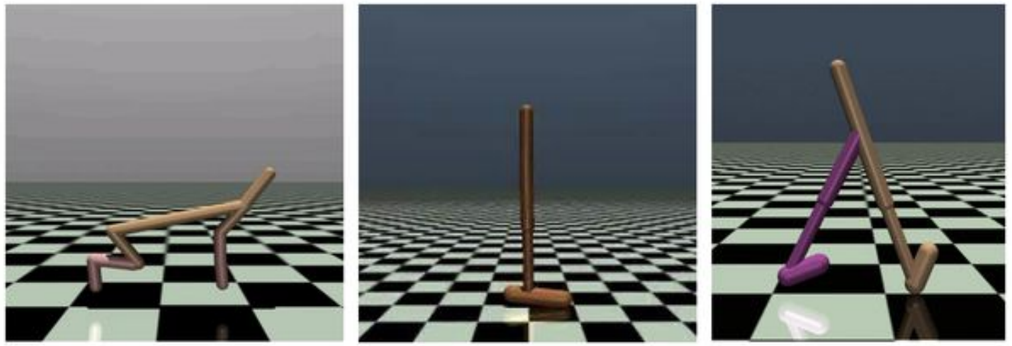
D4RL
(Gym)



Robomimic

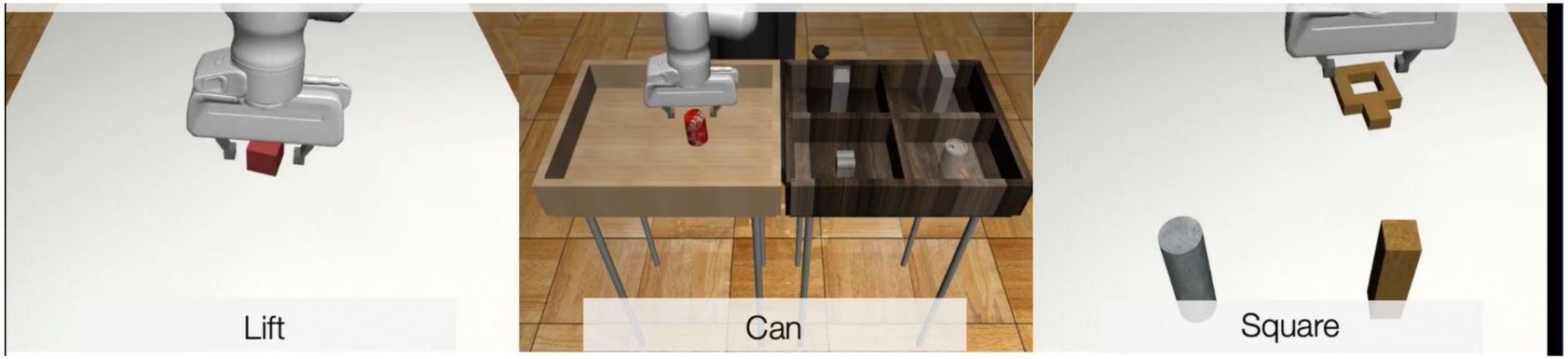


D4RL (Gym)



Halfcheetah Hopper Walker2d

Robomimic



01 How are agents affected by suboptimal data ?

a Sorting the trajectories based on returns to expose the best X% and worst X% of the data to agents.

01 How are agents affected by suboptimal data ?

a Sorting the trajectories based on returns to expose the best X% and worst X% of the data to agents.

b Vary the trajectory lengths

01 How are agents affected by suboptimal data ?

a Sorting the trajectories based on returns to expose the best X% and worst X% of the data to agents.

b Vary the trajectory lengths

c Adding random noise to the data

01 How are agents affected by suboptimal data ?

a Sorting the trajectories based on returns to expose the best X% and worst X% of the data to agents.

b Vary the trajectory lengths

c Adding random noise to the data

02 How do agents perform when the task complexity is increased ?

01 How are agents affected by suboptimal data ?

a Sorting the trajectories based on returns to expose the best X% and worst X% of the data to agents.

b Vary the trajectory lengths

c Adding random noise to the data

02 How do agents perform when the task complexity is increased ?

03 How do agents perform in stochastic environments ?

Baseline Results

Establishing Baseline Results

D4RL

Dataset	DT		CQL		BC
	Sparse	Dense	Sparse	Dense	
medium	**62.56 ± 1.16**	63.66 ± 0.55	43.94 ± 4.7	**67.11 ± 0.24**	53.91 ± 5.93
medium replay	**64.08 ± 1.25**	65.22 ± 1.57	49.04 ± 13.79	**78.41 ± 0.45**	14.6 ± 8.32
medium expert	**103.15 ± 0.77**	103.64 ± 0.12	29.36 ± 5.14	**105.39 ± 0.84**	47.72 ± 5.5
Average	**76.6 ± 1**	77.51 ± 1.12	40.78 ± 7.88	**83.64 ±0.51**	38.74 ± 6.58

Establishing Baseline Results

D4RL

Dataset	DT		CQL		BC
	Sparse	Dense	Sparse	Dense	
medium	**62.56 ± 1.16**	63.66 ± 0.55	43.94 ± 4.7	**67.11 ± 0.24**	53.91 ± 5.93
medium replay	**64.08 ± 1.25**	65.22 ± 1.57	49.04 ± 13.79	**78.41 ± 0.45**	14.6 ± 8.32
medium expert	**103.15 ± 0.77**	103.64 ± 0.12	29.36 ± 5.14	**105.39 ± 0.84**	47.72 ± 5.5
Average	**76.6 ± 1**	77.51 ± 1.12	40.78 ± 7.88	**83.64 ±0.51**	38.74 ± 6.58

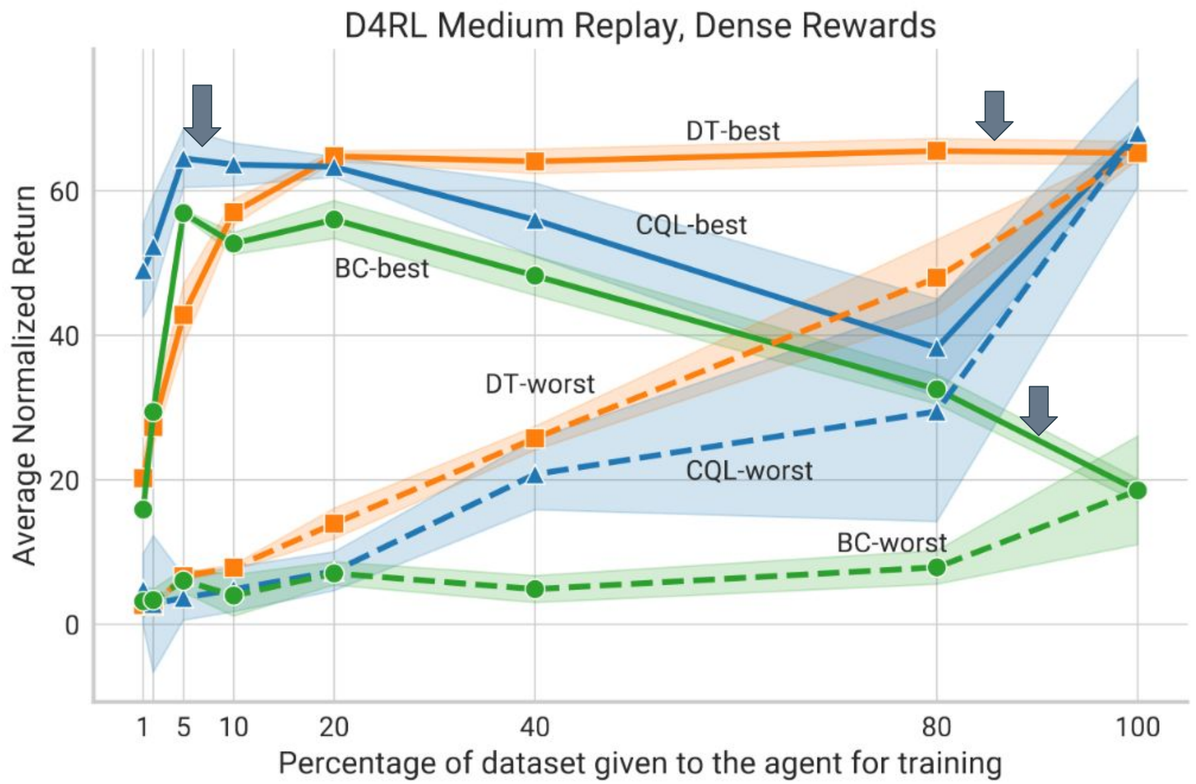
Robomimic
(Machine Generated)

Dataset	DT		CQL		BC
	Sparse	Dense	Sparse	Dense	
Lift	**93.2 ± 3.2**	**96 ± 1.2**	60 ± 13.2	68.4 ± 6.2	59.2 ± 6.19
Can	**83.2 ± 0**	**83.2 ± 1.6**	0 ± 0	2 ± 1.2	55.2 ± 5.8
Average	**88.2 ± 1.6**	**89.6 ± 1.4**	30 ± 6.6	35.2 ± 3.7	57.2 ± 6

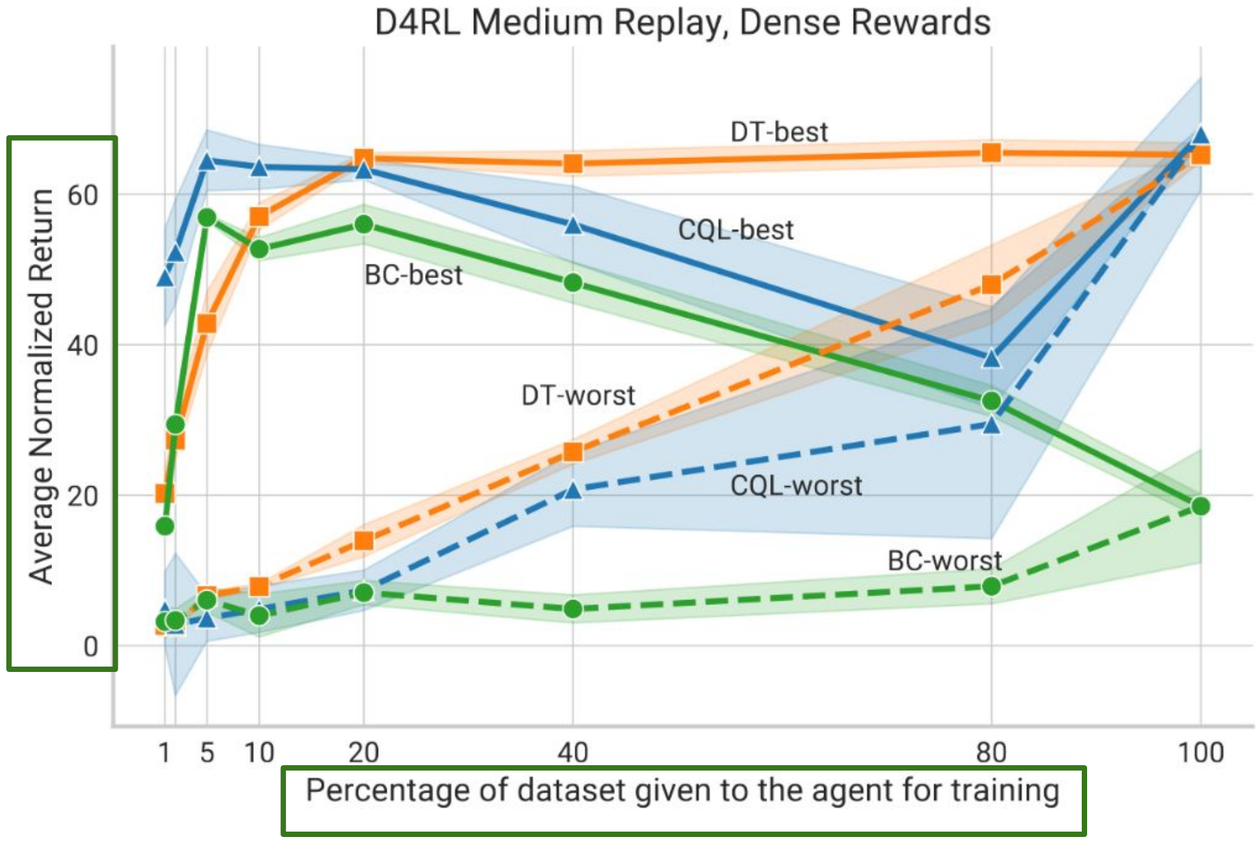
Practical Takeaway: Although CQL excels in certain dense-reward settings, its performance is subject to volatility in other environments. CQL is less preferable for use in sparse-reward settings. Meanwhile, DT is a competitive and risk-averse option that performs well in dense-reward settings and stands out as the top-performing agent in sparse-reward settings.

How are agents affected by suboptimal data ?

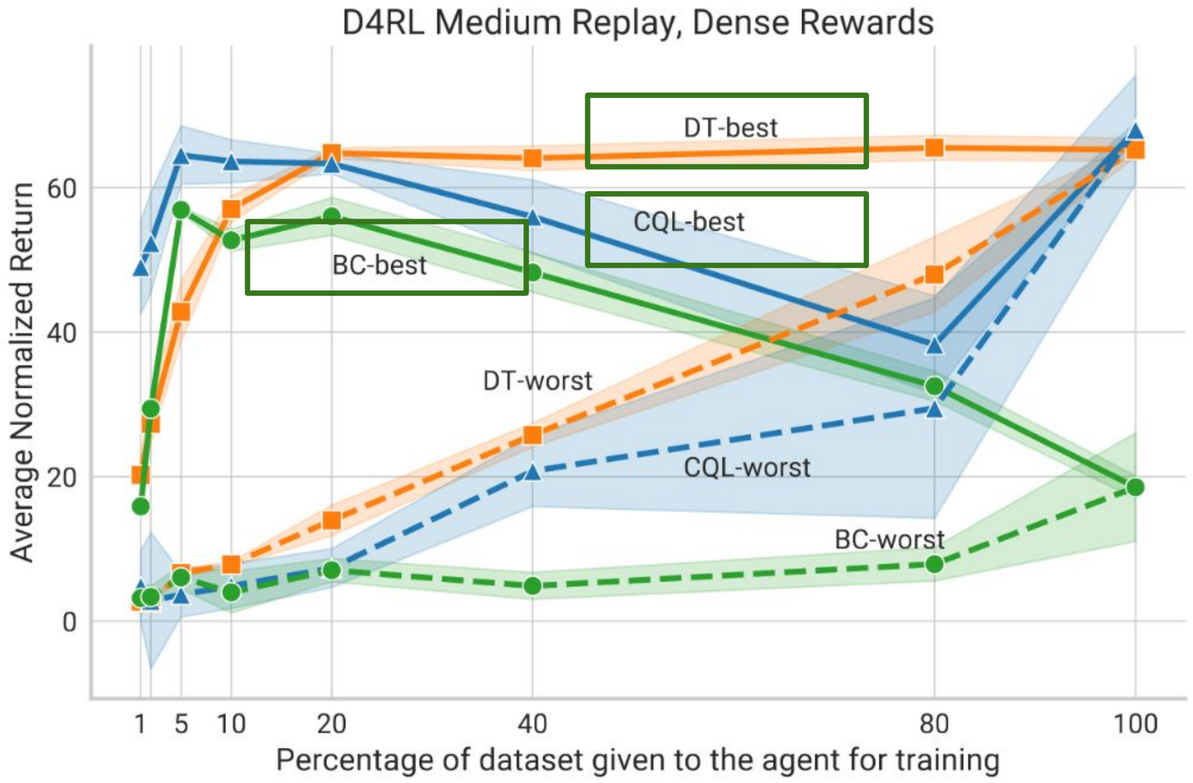
01 How does the amount and quality of data affect each agent's performance?



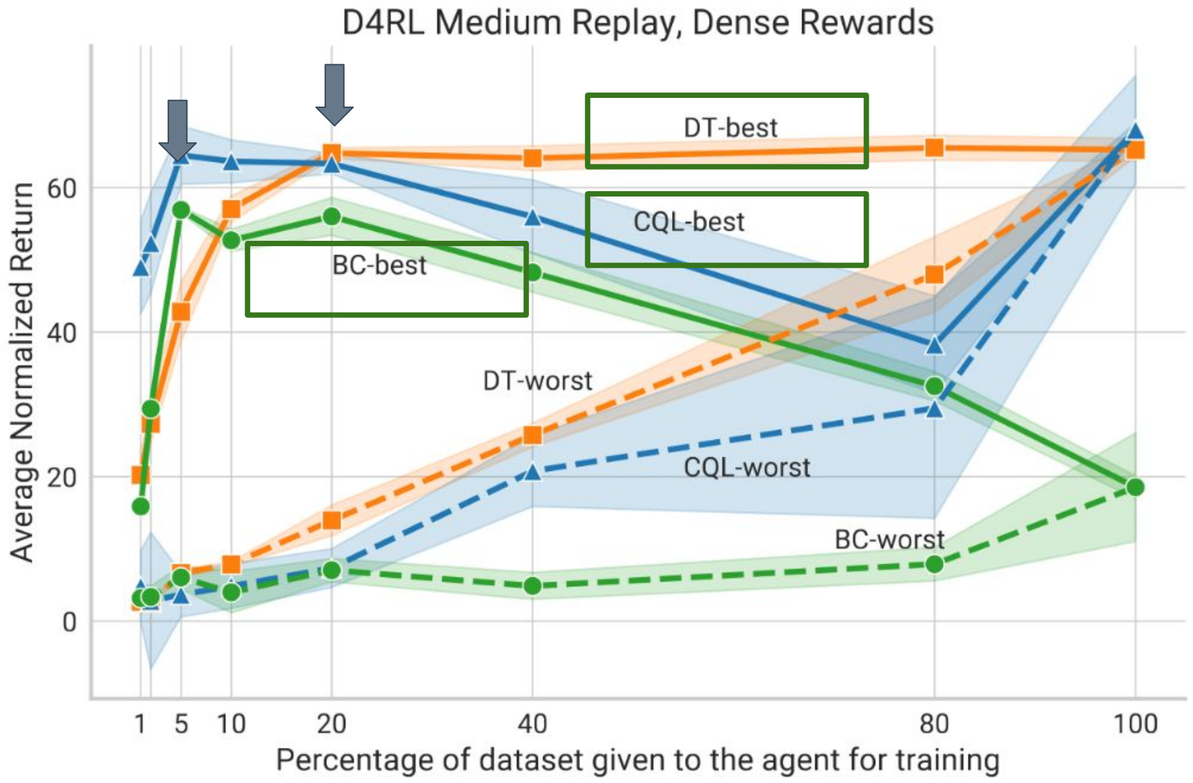
01 How does the amount and quality of data affect each agent's performance?



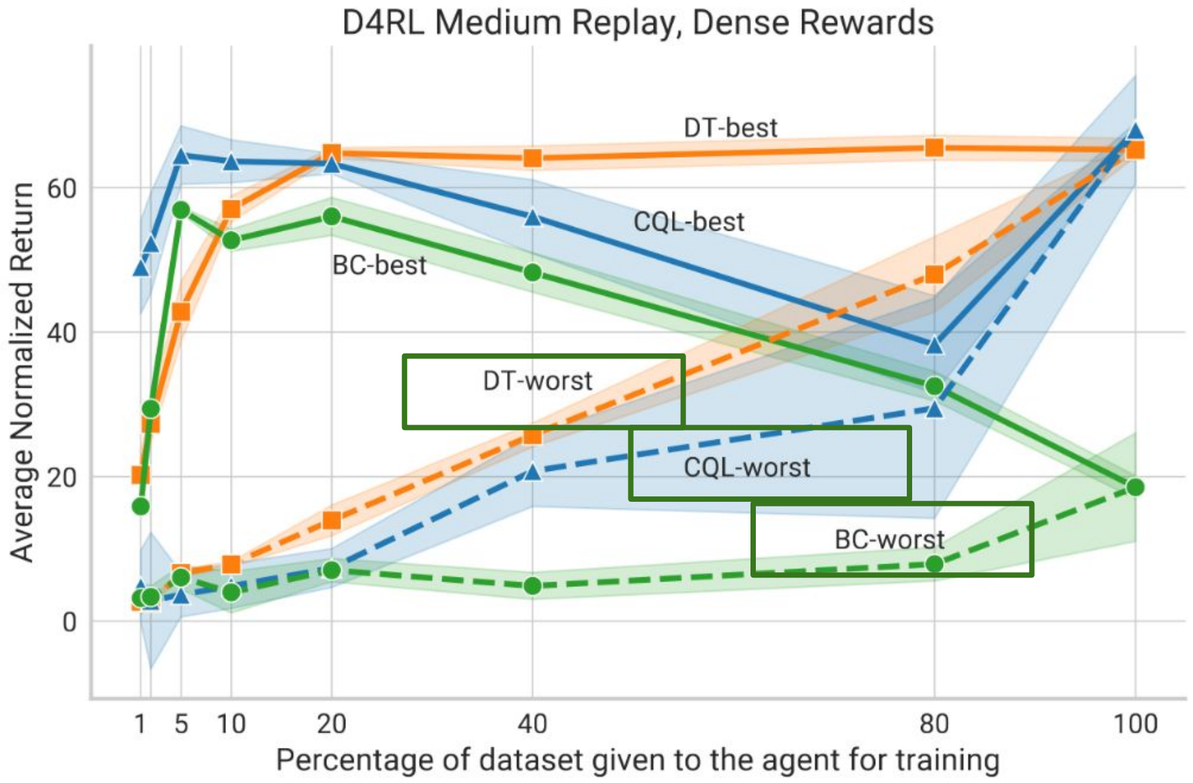
01 How does the amount and quality of data affect each agent's performance?



01 How does the amount and quality of data affect each agent's performance?



01 How does the amount and quality of data affect each agent's performance?

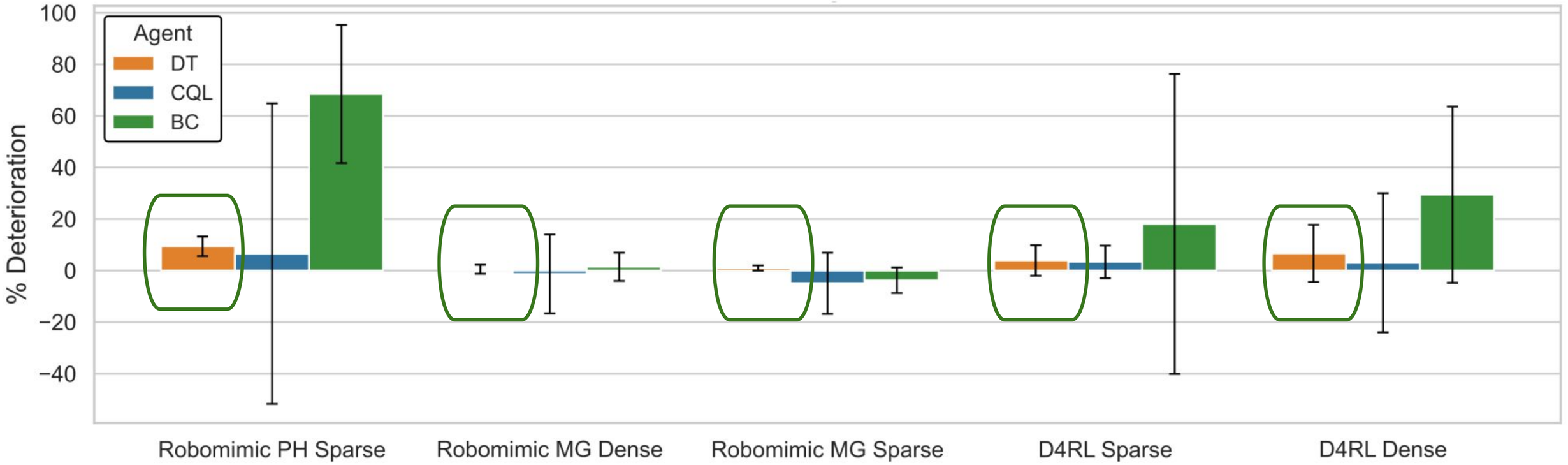


Practical Takeaways: 1) CQL is the most sample-efficient agent given a small amount of high-quality data; 2) While DT requires more data than CQL, it scales more robustly with additional suboptimal data due to DT's reliance on returns-to-go being more robust to variance in rewards; 3) DT can be slightly better than CQL with very low-quality data; 4) DT and CQL are preferable over BC, especially in the presence of suboptimal data.

Dataset Type	Training Trajectory Length	DT	CQL	BC
PH	105 ± 13	83.1 ± 0.8	45.6 ± 5.0	$\mathbf{91.3 \pm 0.9}$
MH-Better	133 ± 33	53.5 ± 0.6	36.5 ± 4.7	$\mathbf{80.2 \pm 2.3}$
MH-Better-Okay	156 ± 50	65.3 ± 1	39.3 ± 5.0	$\mathbf{82.2 \pm 2.3}$
MH-Okay	180 ± 51	54.2 ± 0	29.8 ± 4.9	$\mathbf{65.1 \pm 2.6}$
MH-Better-Okay-Worse	194 ± 93	72.2 ± 1.2	26.5 ± 4.4	$\mathbf{79.6 \pm 3.6}$
MH-Better-Worse	201 ± 107	60.0 ± 0.8	32.4 ± 10.7	$\mathbf{74.2 \pm 3.3}$
MH-Okay-Worse	224 ± 99	52.9 ± 0.5	28.7 ± 2.3	$\mathbf{67.4 \pm 3.4}$
MH-Worse	269 ± 113	52.4 ± 0	5.8 ± 4.1	$\mathbf{59.5 \pm 6.8}$

Practical Takeaway: Agents are affected similarly as trajectory lengths are increased, but when the data was generated by humans, the Imitation Learning paradigm is preferable.

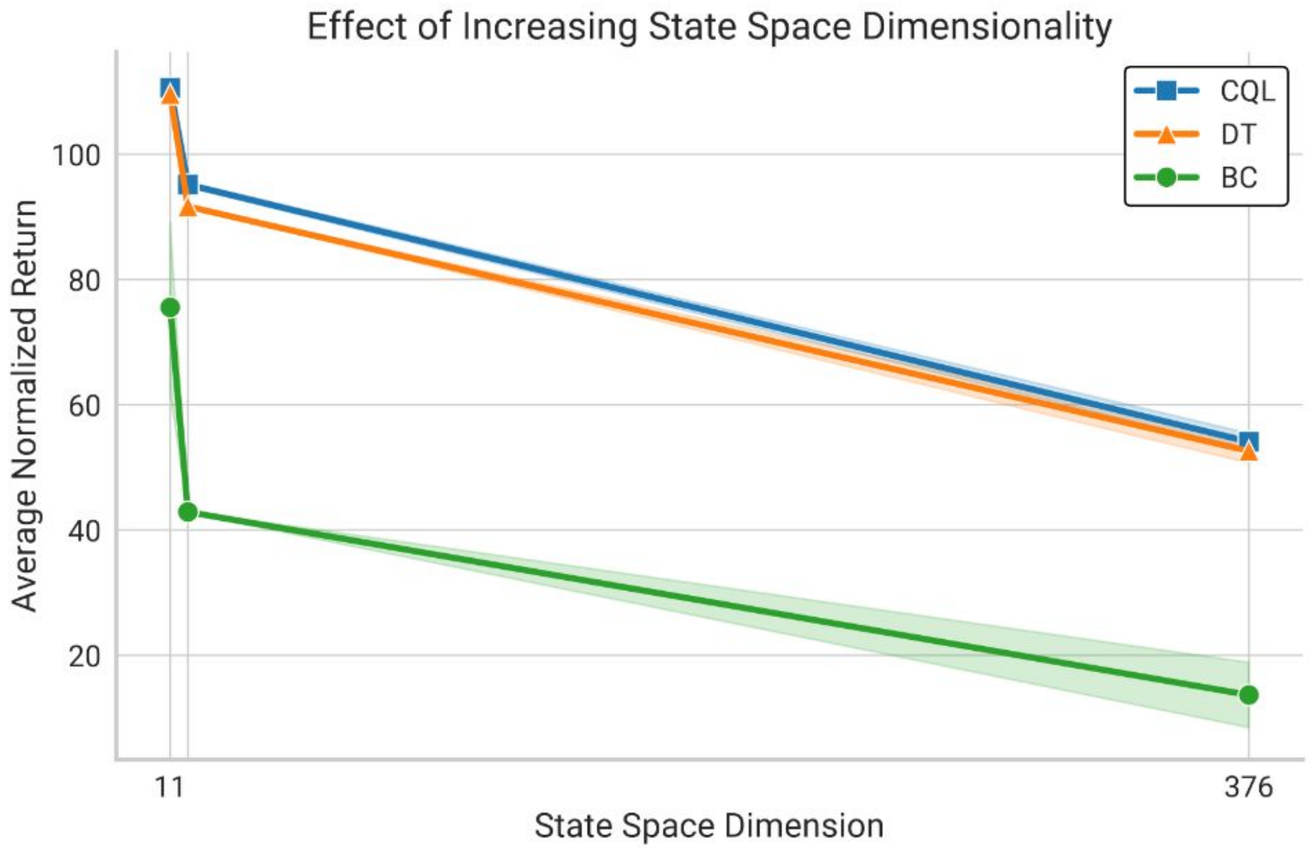
03 How are agents affected when random data is added to the dataset?



Practical Takeaway: BC is likely to suffer the most from the presence of noisy data, while DT and CQL are more robust. However, DT is more reliably performant than CQL in this setting.

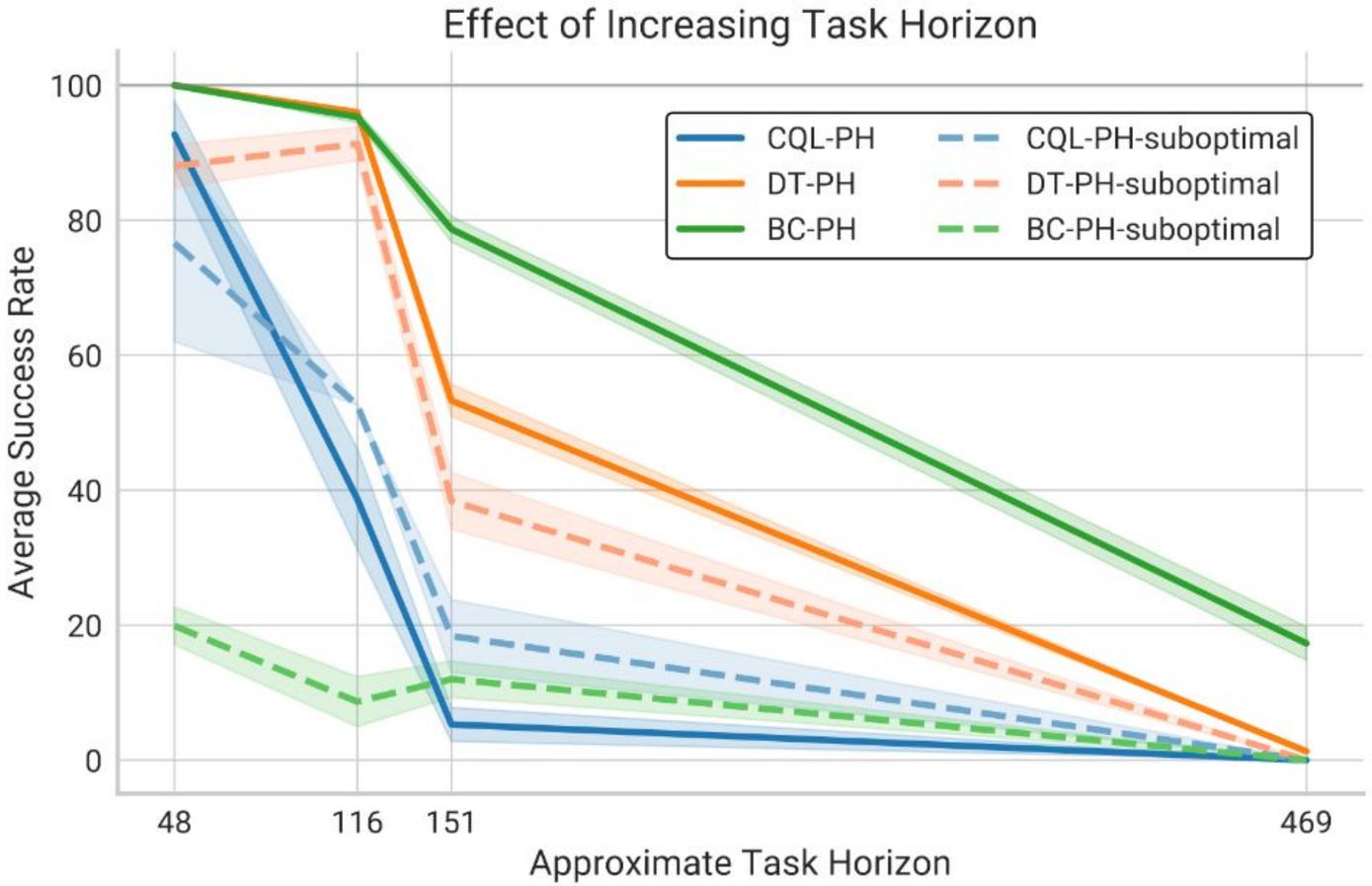
How are agents affected by the complexity of the task ?

How are agents affected by the complexity of the task ?



Practical Takeaway: All agents experience similar deterioration when the dimensionality of the state space is increased. When the task horizon is increased, DT remains a robust choice, but BC may be preferable when data is known to be high-quality.

How are agents affected by the complexity of the task ?

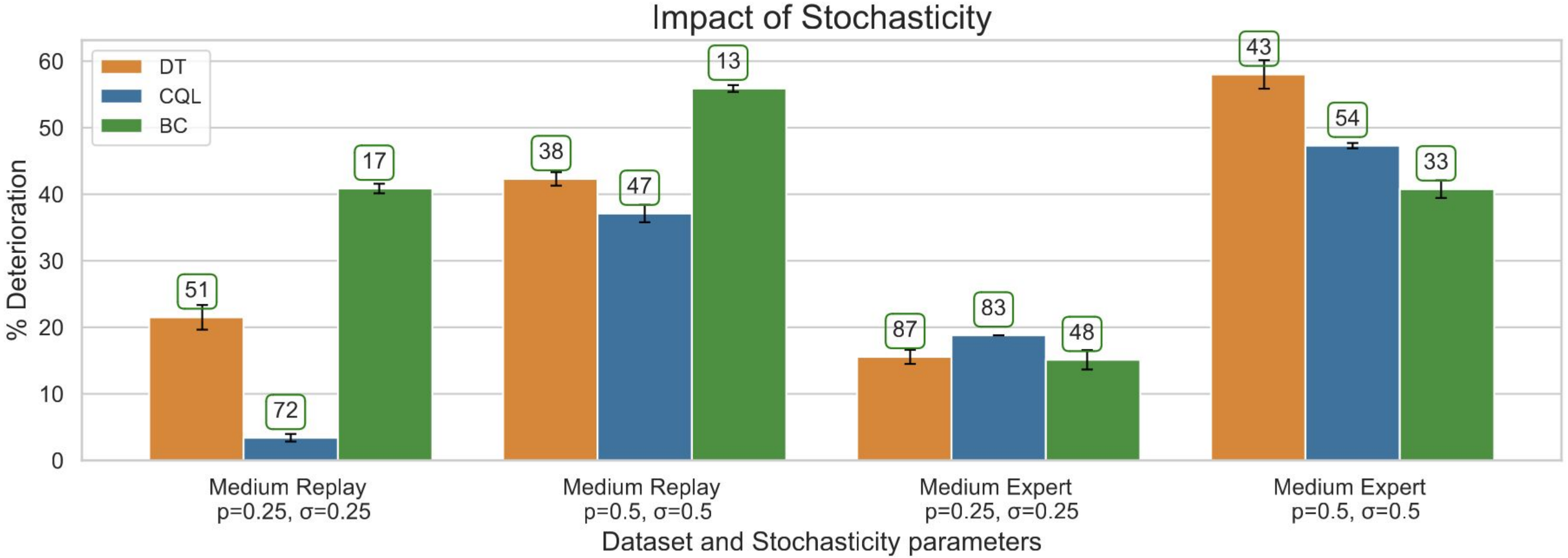


Practical Takeaway: All agents experience similar deterioration when the dimensionality of the state space is increased. When the task horizon is increased, DT remains a robust choice, but BC may be preferable when data is known to be high-quality.

How are agents behave in stochastic environments ?

How do agents behave in stochastic environments ?

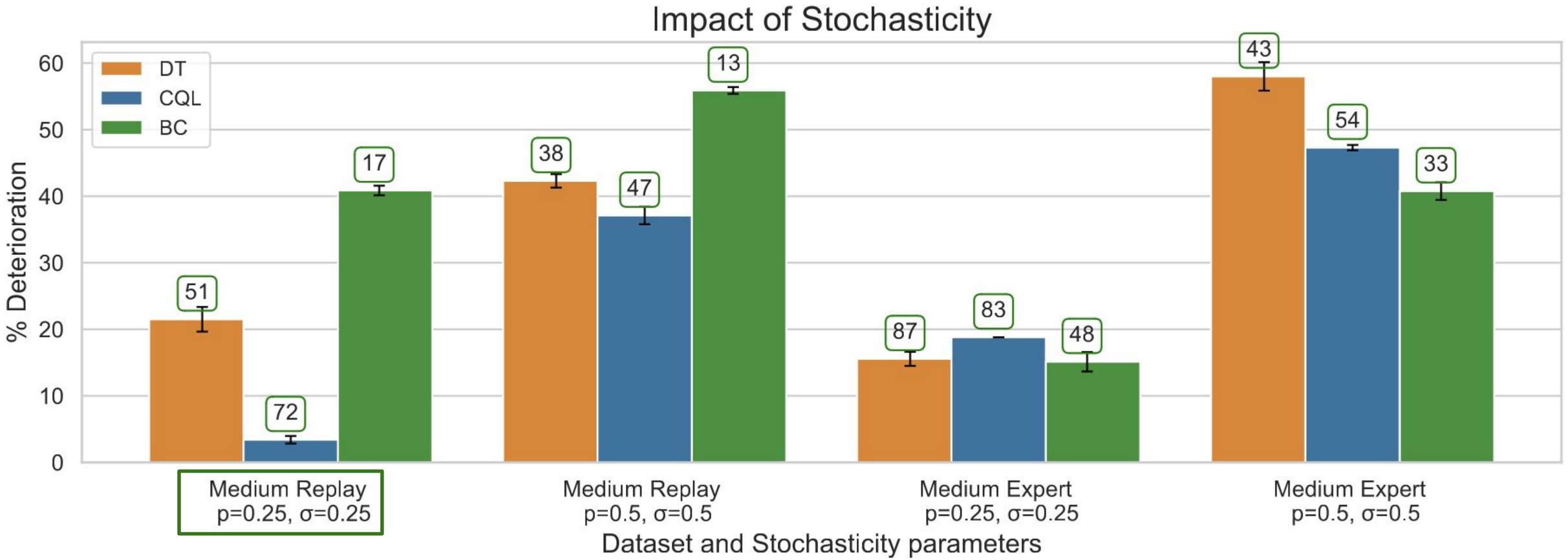
$$action = action + (\mathcal{N}(0,1) * \sigma + \mu)$$



Practical Takeaway: While DT exhibits a comparable decline to CQL when trained on high-quality data in continuous action spaces, CQL is expected to be relatively more robust as data quality declines or stochasticity increases.

How do agents behave in stochastic environments ?

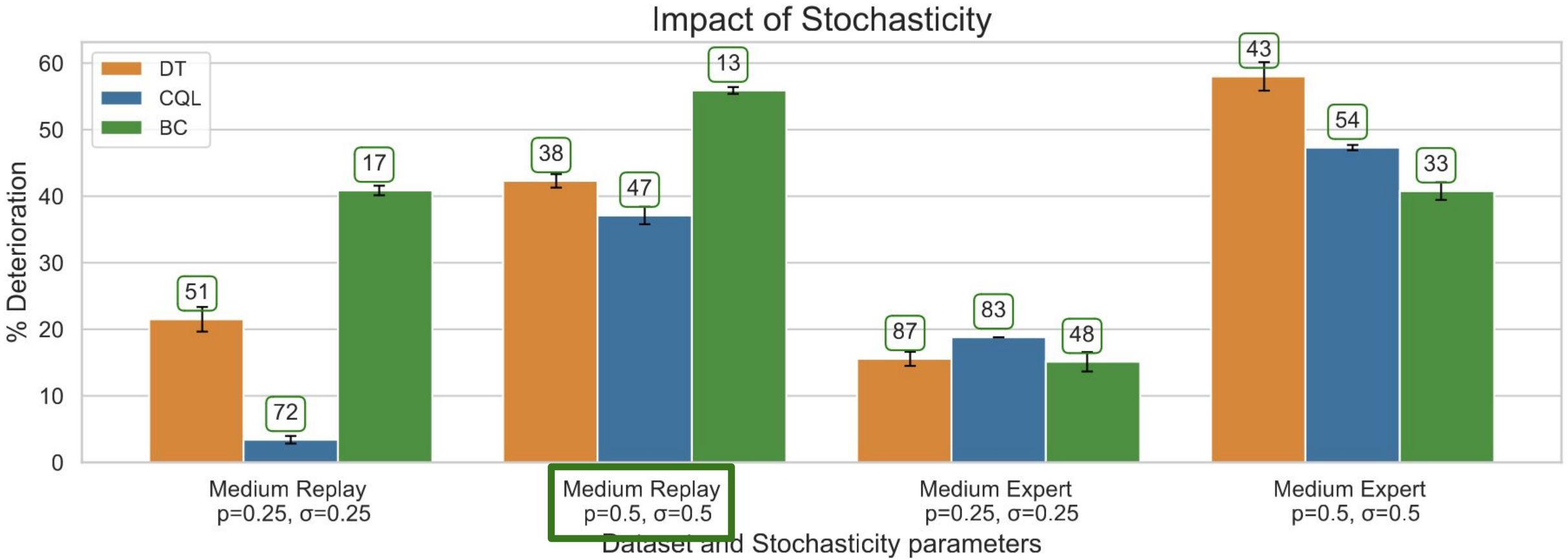
$$action = action + (\mathcal{N}(0,1) * \sigma + \mu)$$



Practical Takeaway: While DT exhibits a comparable decline to CQL when trained on high-quality data in continuous action spaces, CQL is expected to be relatively more robust as data quality declines or stochasticity increases.

How do agents behave in stochastic environments ?

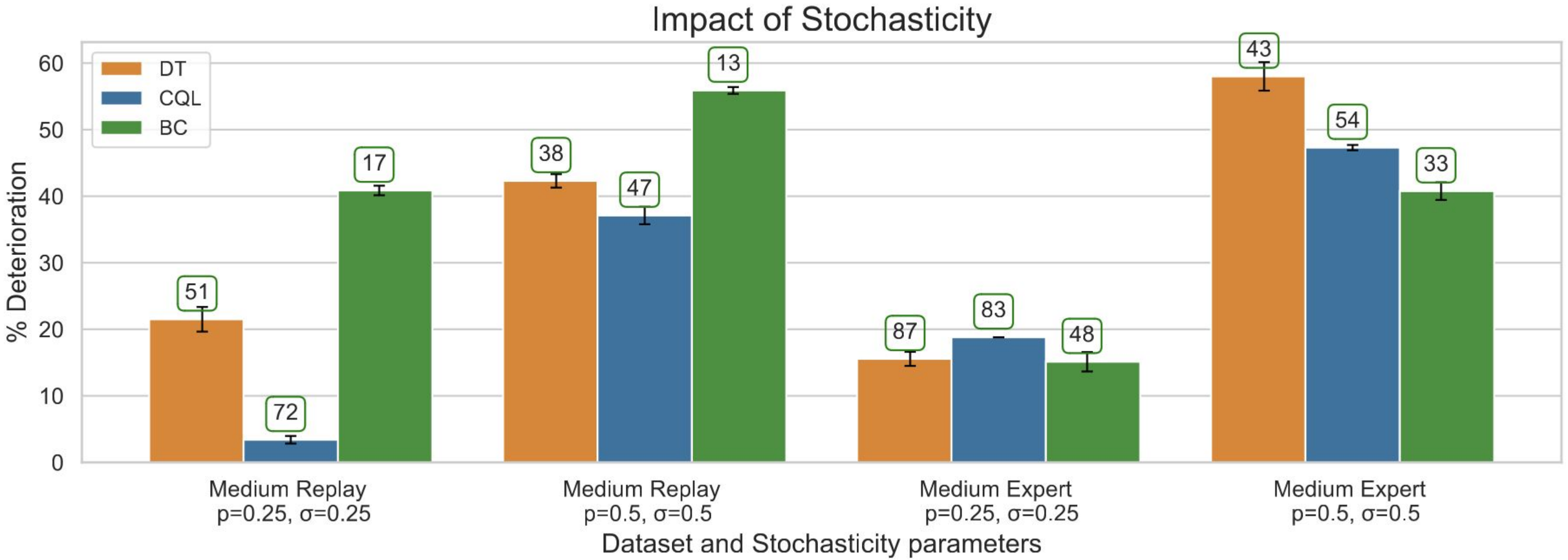
$$action = action + (\mathcal{N}(0,1) * \sigma + \mu)$$



Practical Takeaway: While DT exhibits a comparable decline to CQL when trained on high-quality data in continuous action spaces, CQL is expected to be relatively more robust as data quality declines or stochasticity increases.

How do agents behave in stochastic environments ?

$$action = action + (\mathcal{N}(0,1) * \sigma + \mu)$$



Practical Takeaway: While DT exhibits a comparable decline to CQL when trained on high-quality data in continuous action spaces, CQL is expected to be relatively more robust as data quality declines or stochasticity increases.

Conclusion